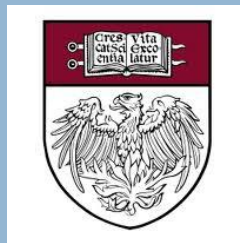# An Evaluation of Difference and Threshold Techniques for Efficient Checkpoints

Sean Hogan, Large Scale Systems Group, University of Chicago

Jeff R. Hammond

Andrew A. Chien

FTXS 2012 Workshop
June 25th, 2012, Boston, MA

Outline

- Motivation

- Traditional Checkpointing Model

- Differenced Checkpointing

- Differenced Checkpointing with Threshold

- Related Work

- Summary and Future Work

The Problem

- Checkpointing – widely used technique
- Current checkpointing costs as high as 10% of system time
- Technology trends
  - Increased rate of bit errors, power failures, hardware failures
  - Lower I/O to compute ratio
    - [2011 CCC Study]
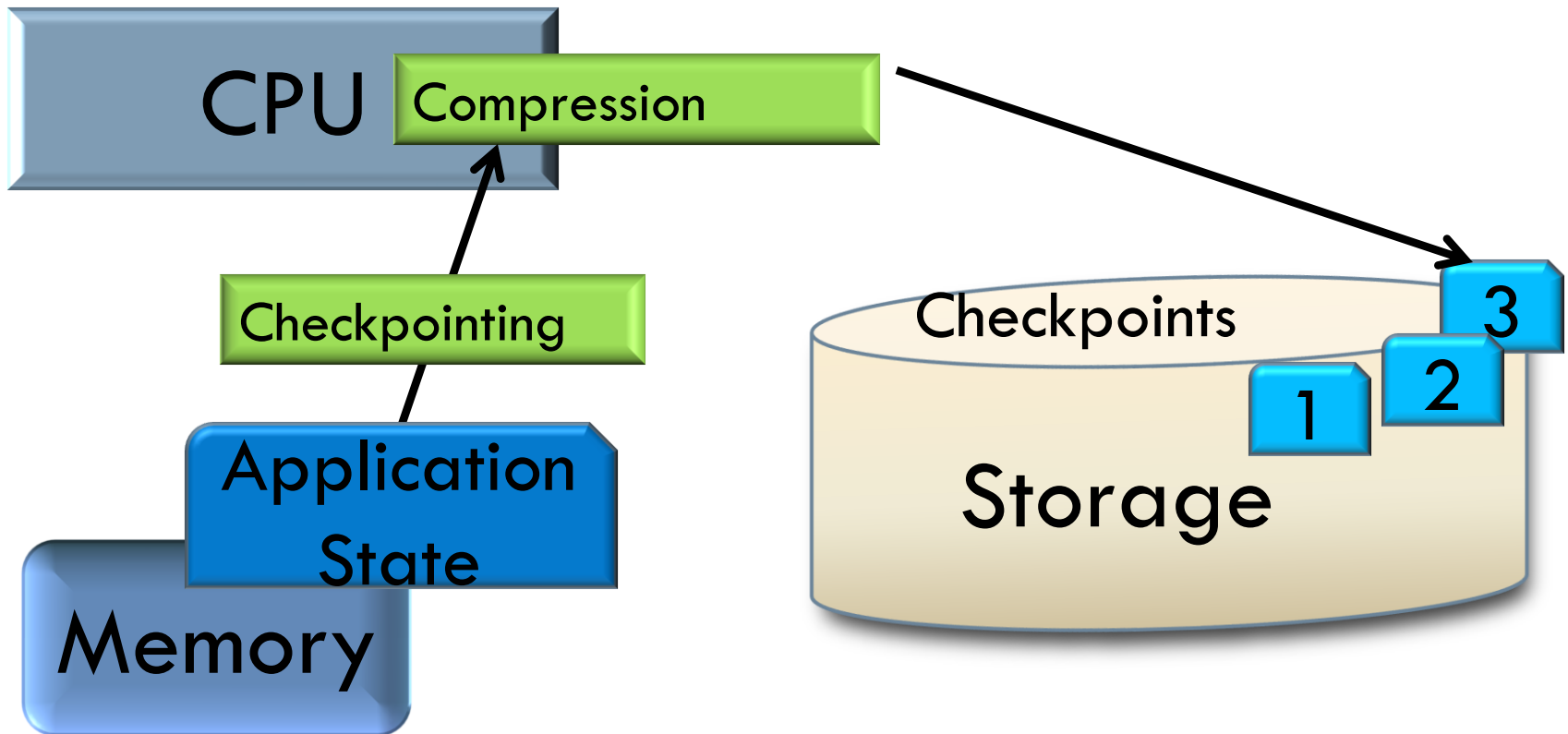- By 2020 these challenges threaten viability of large scale systems

Approach

- Goal: Reduce cost of checkpointing

  - Reduce time

  - Reduce size

- Evaluate three methods of checkpointing

  - Compressed

  - Compressed differences

  - Compressed differences with thresholding

Traditional Checkpointing

CPU

Application State

Memory

Checkpointing

Checkpoints

3

2

1

Storage

FTXS 2012 Hogan
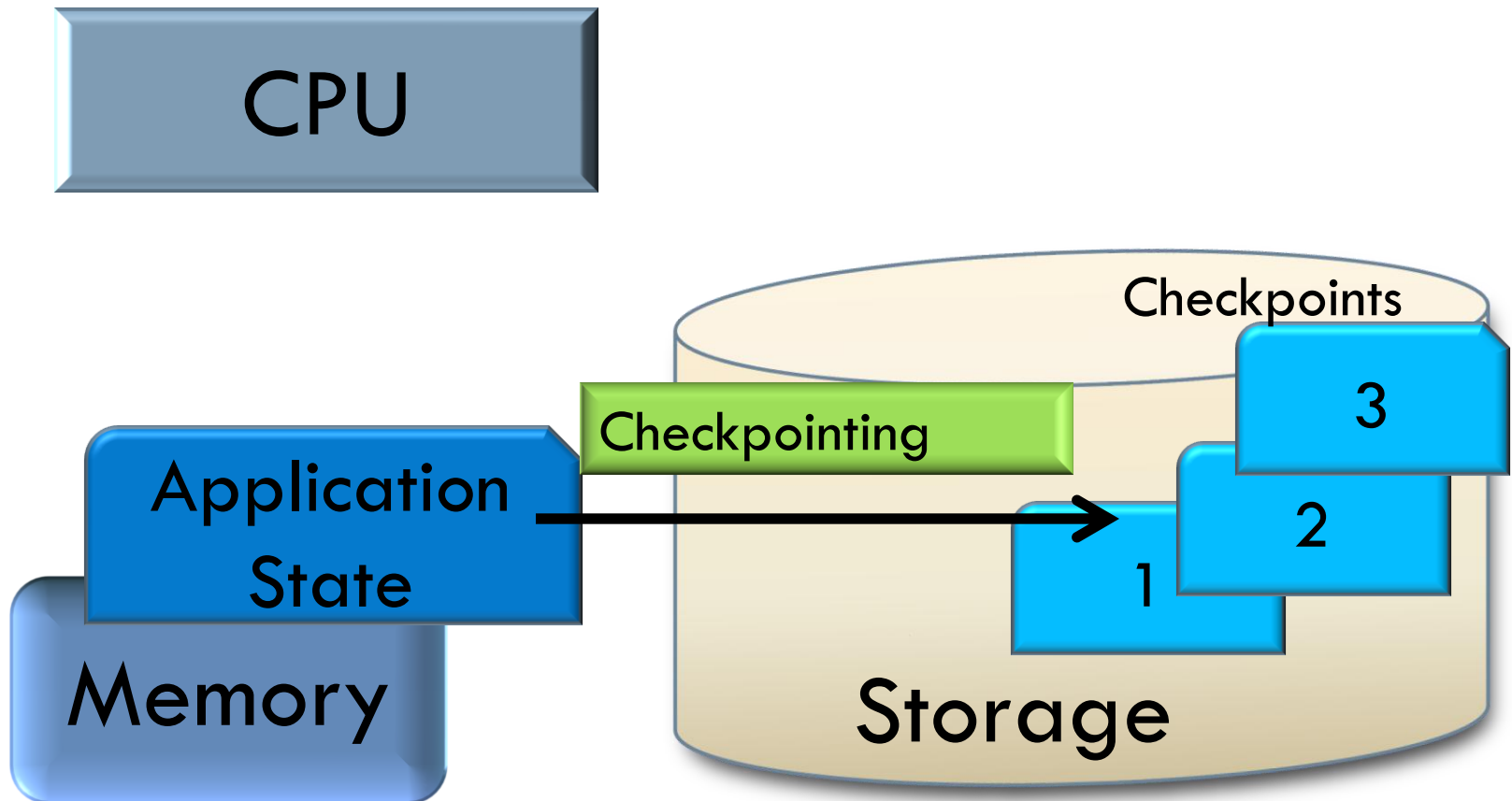
Compressed Checkpoints

CPU

Compression

Checkpointing

Application State

Memory

Checkpoints

3

2

1

Storage

Experimental Background

- FPC (Floating point compressor) [Burtscher, 2009]
  - Domain-specific (64-bit FP Data),
  - Constant-time
  - Based on value prediction
- DEFLATE (LZ77 + Huffman Encoding) [Lempel-Ziv, 1977]
  - General purpose
  - Variable run-time (based on parameters)
  - Exploits sub-string patterns
- NWChem: Computational Chemistry
  - 3 run sizes: w3, w4, w13 (45MB to 328 MB per checkpoint)
  - Coupled-cluster method, simulates systems of water clusters
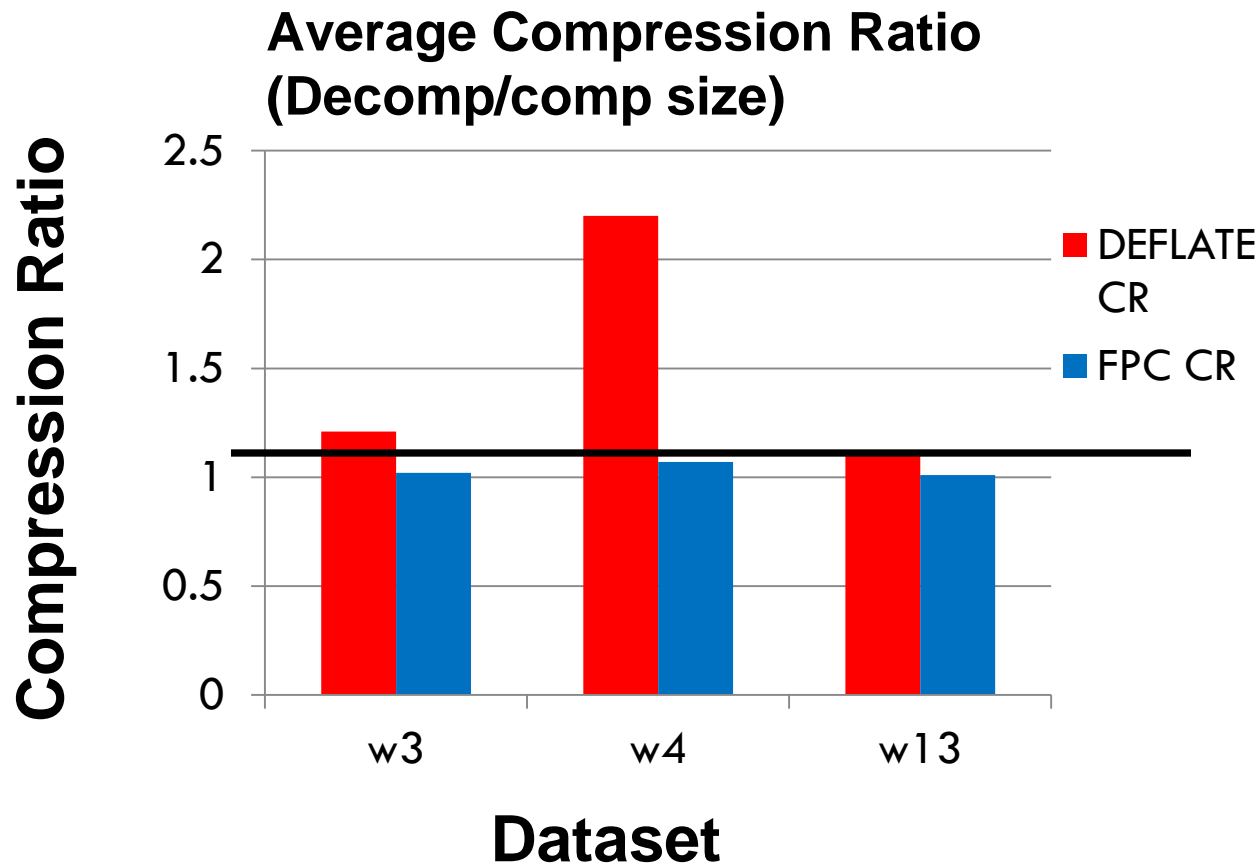
Experimental Background

- Compressing single checkpoints

- Determine base difficulty of reducing checkpoint cost

Compression Throughput

Average throughput
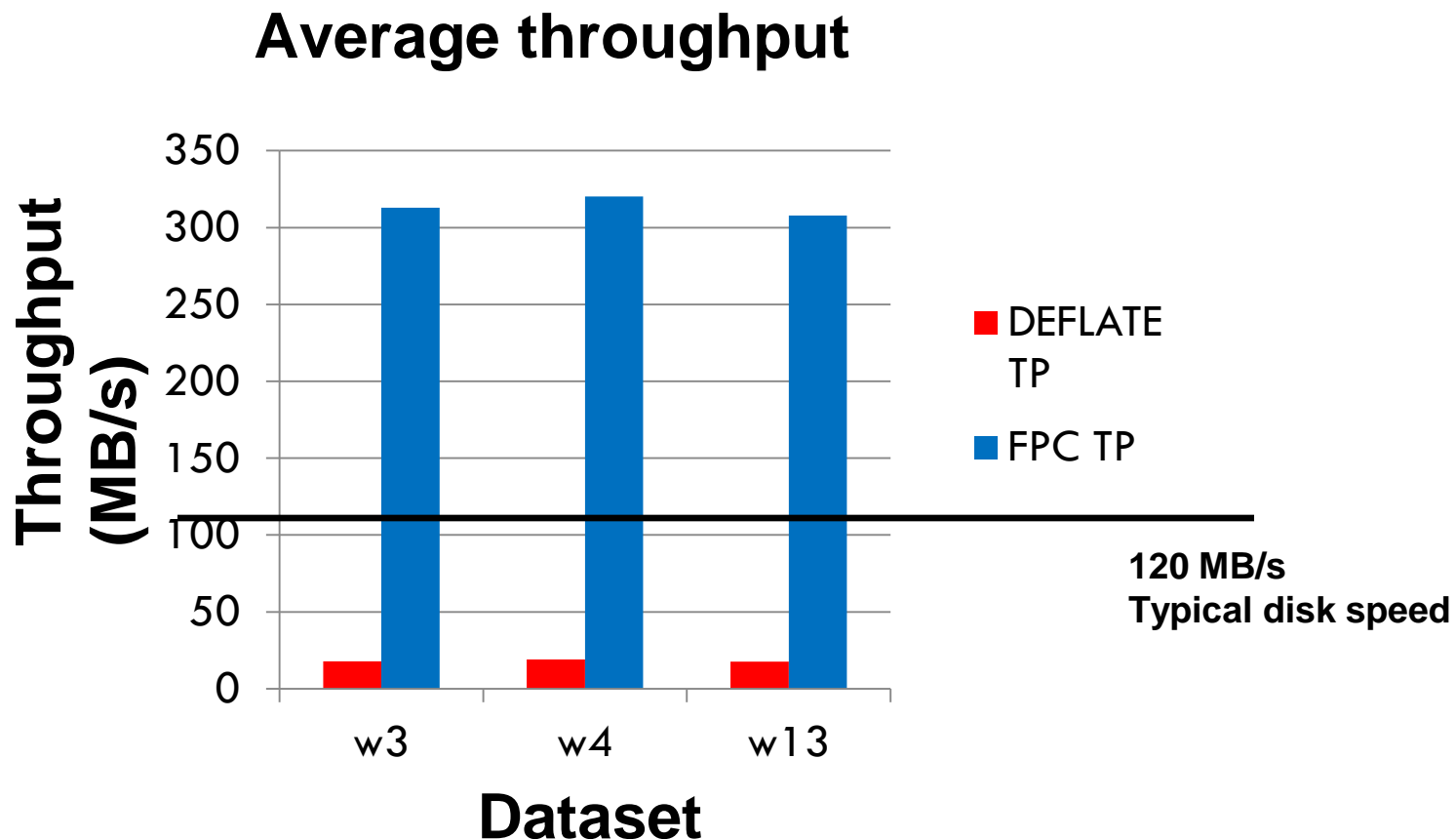
- FPC faster than DEFLATE

Compression Ratio
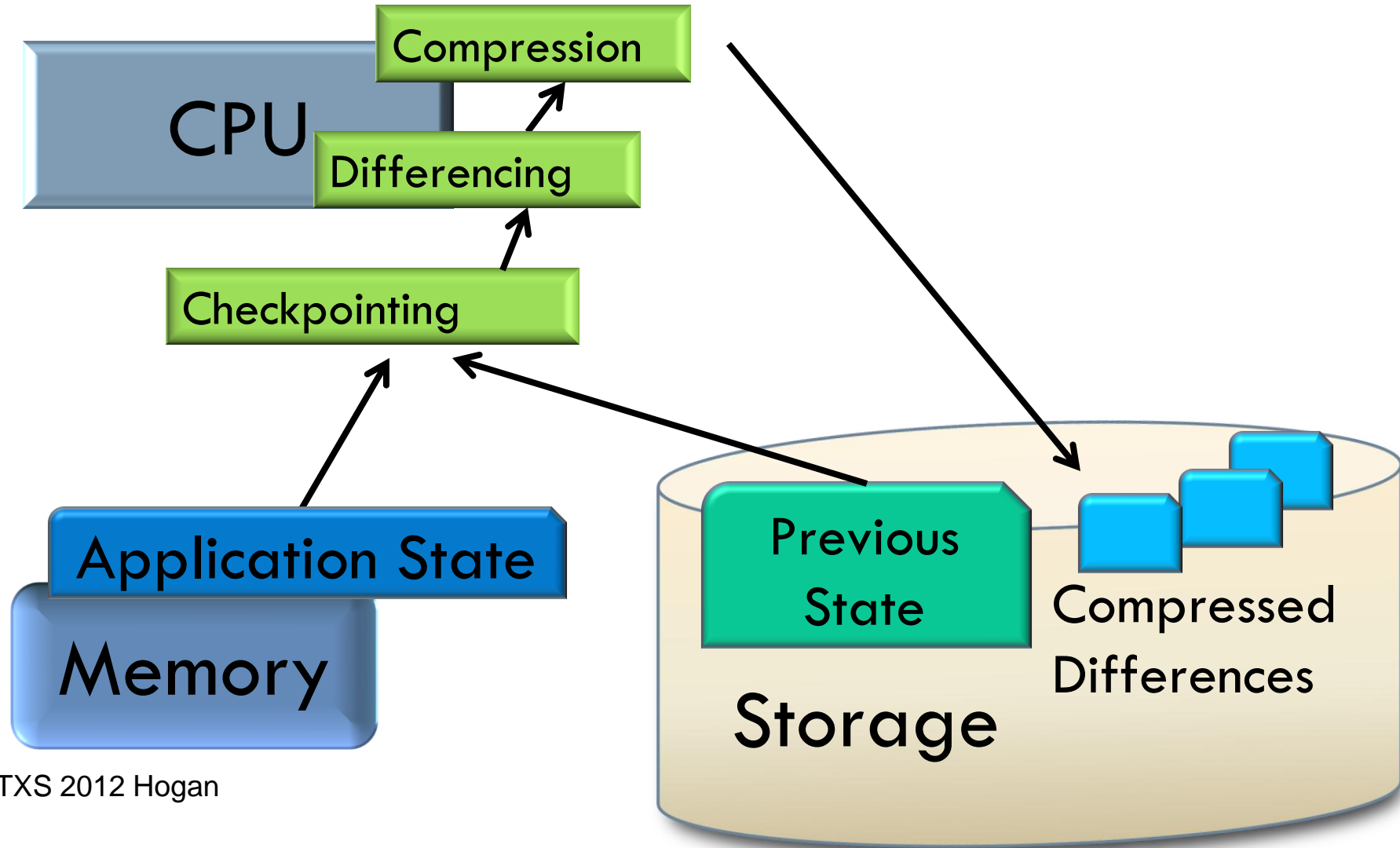

Average Compression Ratio (Decomp/comp size)

- Floating point data difficult to compress

Method #1: Compressed Differences

- Computations have an evolution of values

- Checkpoint differences have smaller magnitude and fewer significant bits than raw checkpoints

- Idea: Try to compress checkpoint differences in context of application values

  - Example: value changes from 1.00 to 1.01, compress a representation of 0.01 as a delta to 1.00
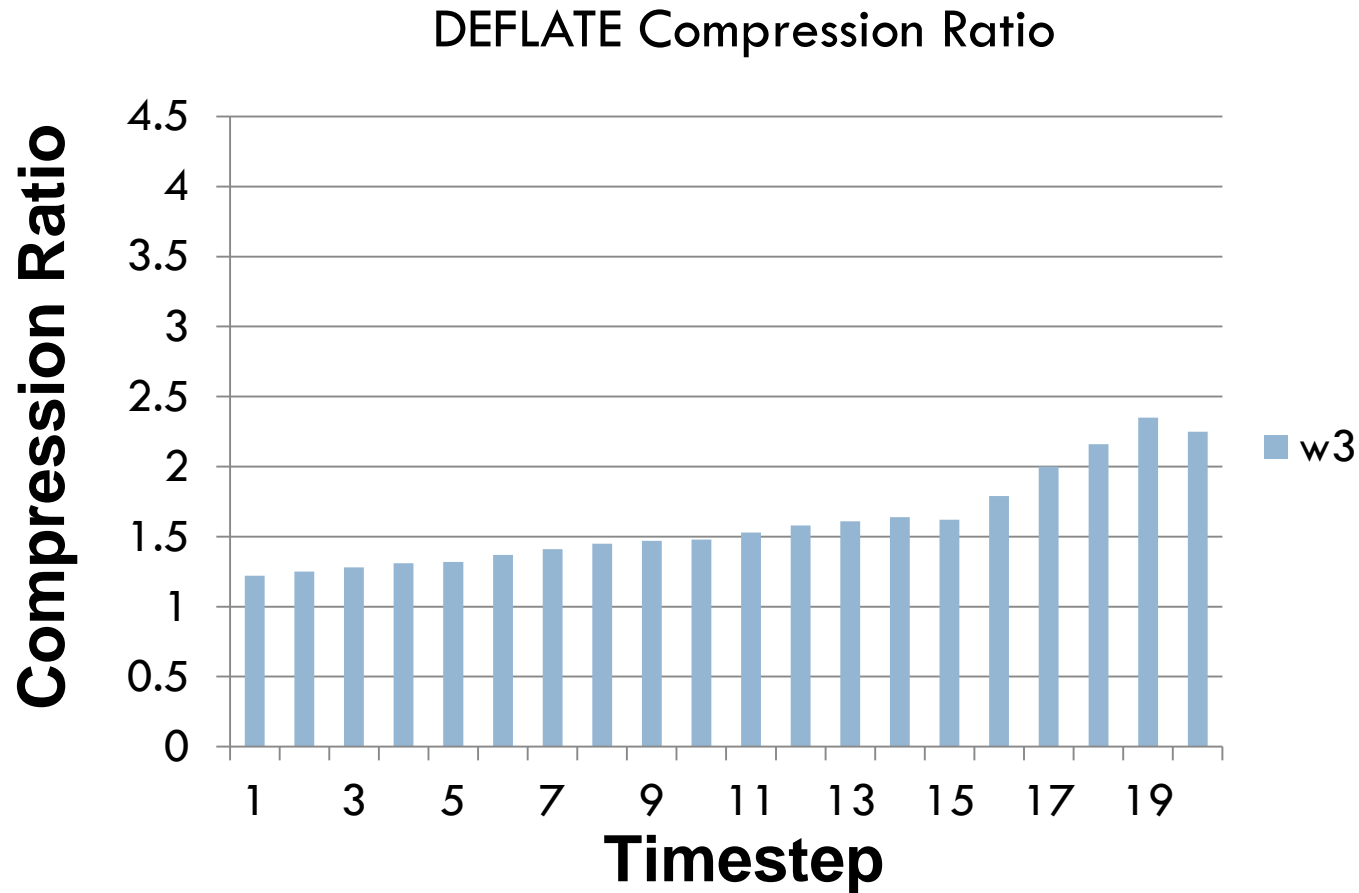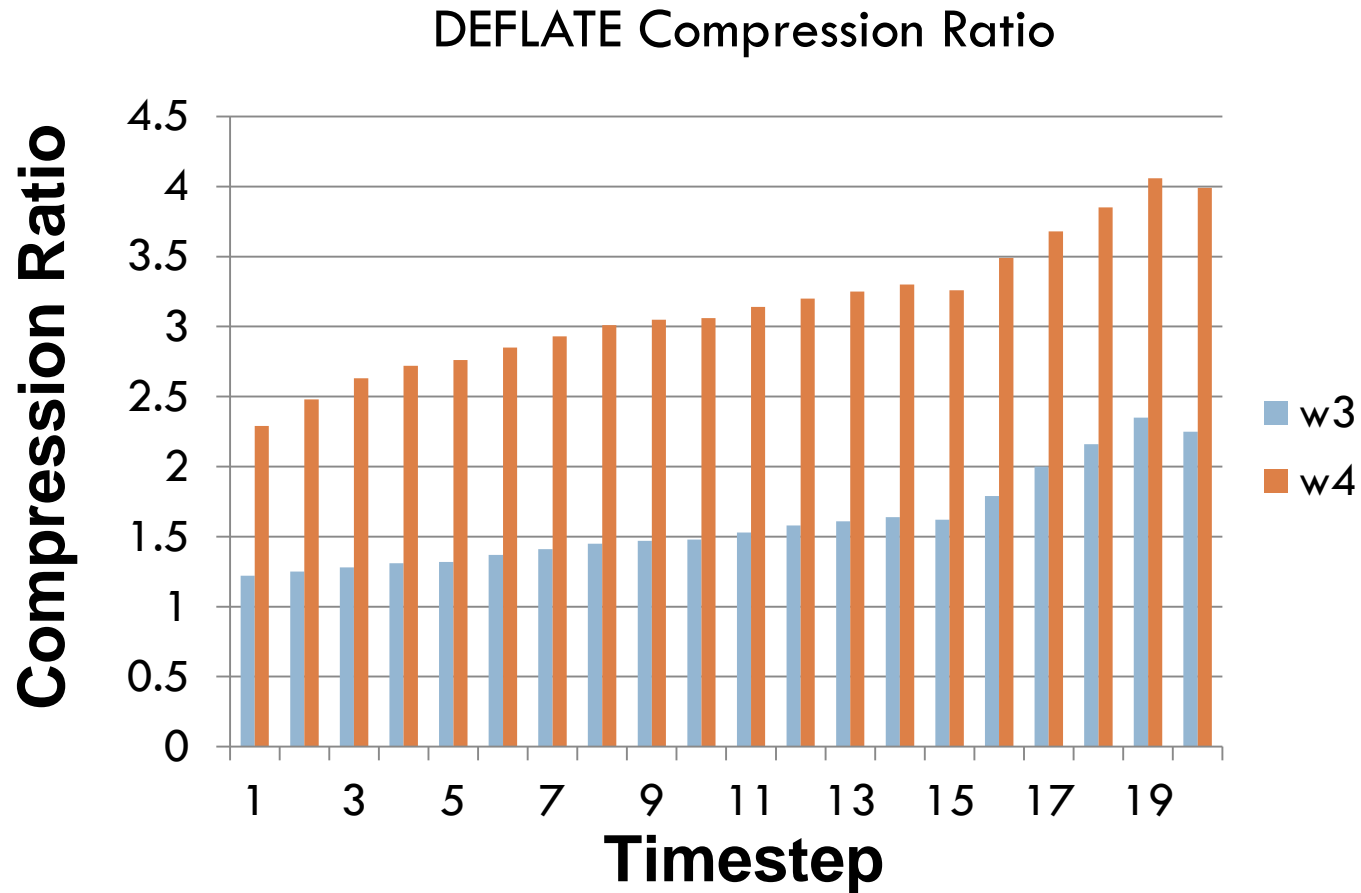
Method #1: Compressed Differences

Method #1: Compressed Differences

- What changes between compression of checkpoints and their differences?

- Look at sets of checkpoints from a computation

- Perform differencing + compression on successive pairs of checkpoints

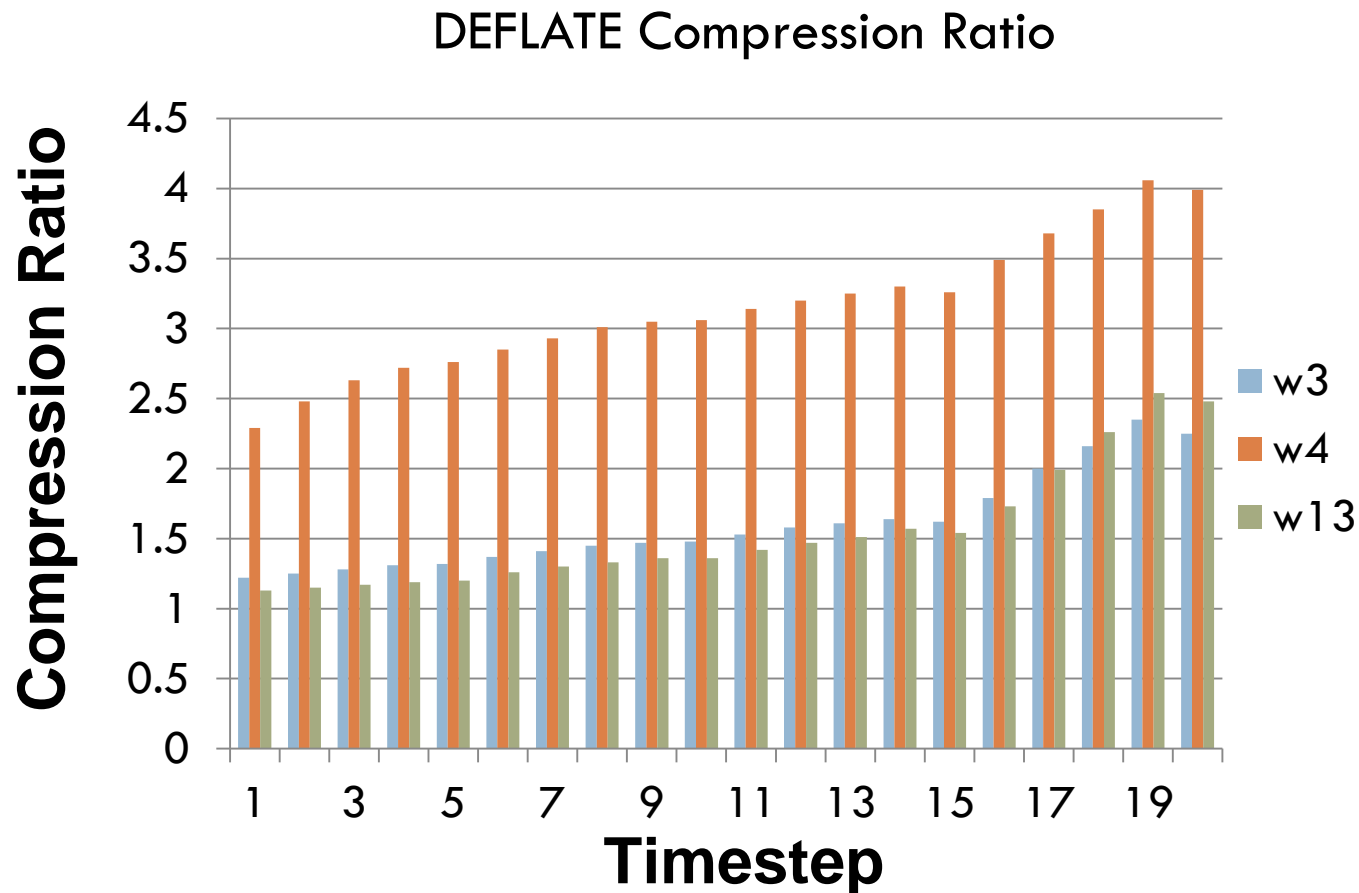# Method #1: Compressed Differences

DEFLATE Compression Ratio

Method #1: Compressed Differences

DEFLATE Compression Ratio

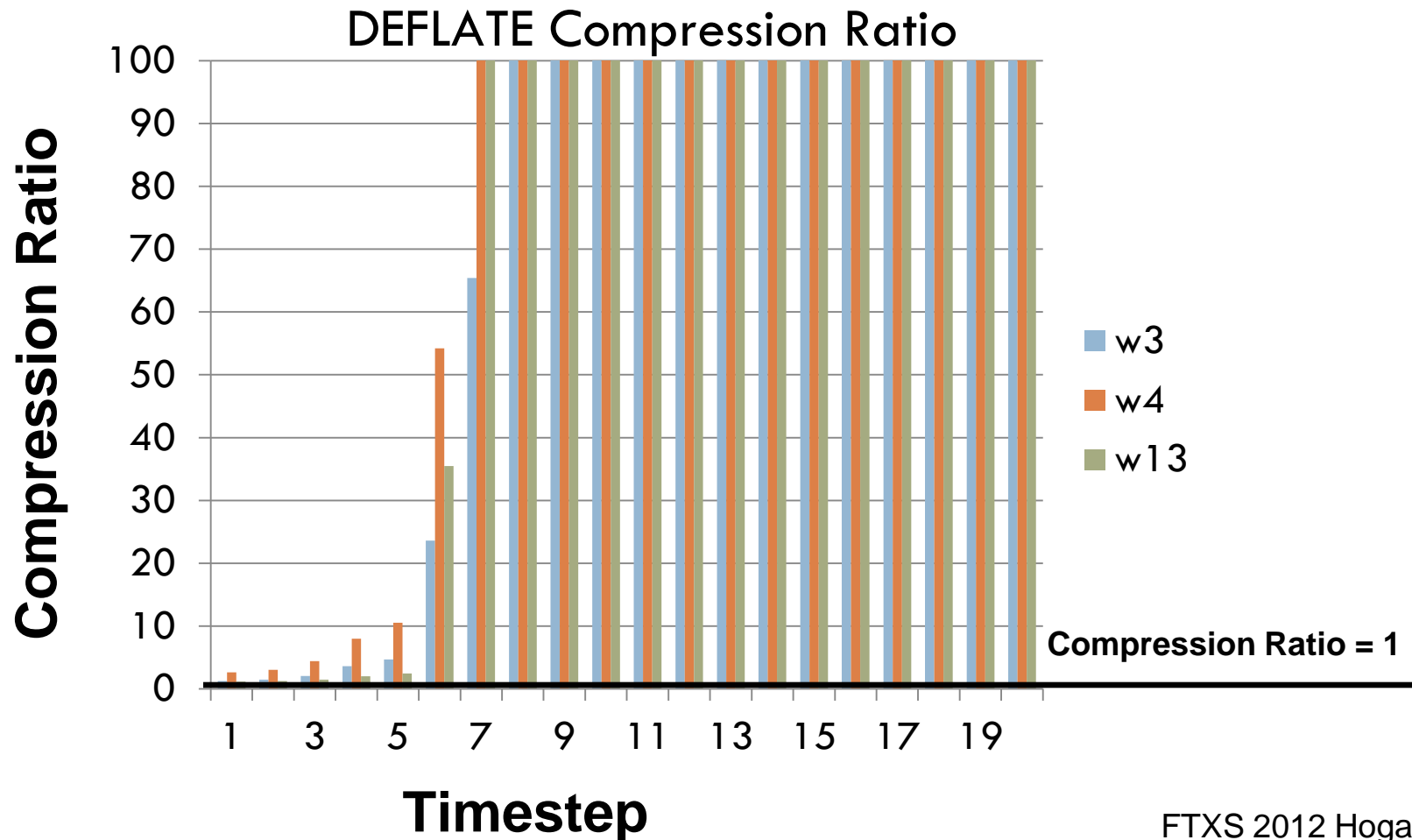# Method #1: Compressed Differences

DEFLATE Compression Ratio

- Taking advantage of application can increase compression

Method #2:
Compressed Differences with Threshold

- How does rounding values below a given threshold to zero affect the ability to compress a differenced checkpoint?

- Set a threshold value

- Differencing and compression on successive pairs of checkpoints, with a 10^{-7} cutoff

Method #2:
Compressed Differences with Threshold

DEFLATE Compression Ratio

Legend: w3, w4, w13

Compression Ratio = 1

X-axis: Timestep

Y-axis: Compression Ratio

FTXS 2012 Hogan

- Quick convergence in single precision

Related Work

- Exploiting application and floating point structure

  - GPU-driven Compression (O'Neil 2011)

  - Predictor-based compression (Burtscher 2009)

  - Data pre-conditioning (Schendel 2012)

- System techniques

  - Protocols for uncoordinated checkpointing (Guermouche 2011)

  - Coordinated checkpointing method evaluations (Buntinas 2007)

  - Dynamically changing checkpointing methods (Moody 2010)

- Failure structure of alternate storage

  - Memory and SSD-focused checkpointing(Gomez 2010)

Conclusions

- Increasing costs of checkpointing are a critical challenge

- Changes in dataset can be more compactly represented than the dataset itself

  - Application-based thresholding increases compression

- Convergence of differences of application state is exploitable: 2.5 – 4.0 compression ratio on differences

- Application-based numerical precision requirements are exploitable: > 1000.0 compression ratio for thresholding

- Changing the precision of a checkpoint through a computation can increase efficiency

Future Work

- Assess recovery time of differenced checkpoints

- Broader experiments – larger systems, more applications

- Exploit application structure further with calculated and varying thresholds

- Exploration of different compression algorithms

FTXS 2012 Hogan

Questions

- seanhogan@uchicago.edu

- jhammond@alcf.anl.gov

- achien@cs.uchicago.edu

- Code and scripts:
github.com/SeanHogan/lssg/tree/master/nwchemtesting/ga

  - ga-delta.c